一种新的基于 H.264/AVC 的可逆鲁棒隐写方法 *

刘舒扬 [1]，陈　亮 [2]

(1. 兰州大学，兰州 730000; 2. 国家计算机网络应急技术处理协调中心，北京 100029)

摘　要：提出了一种新的基于 H。264/AVC 的可逆鲁棒视频隐写方法。该方法首先利用（t，n）门限秘密共享技术将需要嵌入的秘密信息用多项式分发获得子秘密，再利用 BCH 码对子秘密进行编码，然后通过 H。264 的预测模式选择嵌入位置，最后将编码后的子秘密嵌入到视频的各个帧中。提取时，利用视频中 t 个编码后的子秘密恢复秘密信息。实验结果标明，该方法在提取秘密信息时能完全恢复原视频，具有较强的比特错、帧错鲁棒性，在随机丢帧率小于等于 15%的情况下能完全恢复秘密信息，能效控制帧内失真漂移，视觉效果良好。

关键词：H；264/AVC；可逆隐写；鲁棒性；帧内失真漂移

中图分类号：TP309.2　　doi: 10.3969/j.issn.1001-3695.2017.11.0999

New robust reversible steganography method in H. 264 / AVC

Liu Shuyang[1], Chen Liang[2]

(1. *Lanzhou University*, *Lanzhou* 730000, *China*; 2. *National Computer Network Emergency Response Technical Team/Coordination Center of China*, *Beijing* 100029, *China*)

Abstract: This paper proposed a new reversible robust steganography method in H. 264 / AVC. It first distributed the embedded secret data by a specific polynomial and obtain a series of sub-secrets. Then it used the BCH code to encode each group of the sub-secrets and embed the encoded sub-secrets into each frame of the video with prediction mode. And it recovered the hidden data by using the encoded t sub-secrets of the video frames. The experiment results shown that the method can recover the original video completely and has the high bit error and frame error robustness. When the random frame rate is less than or equal to 15%, the method can recover the secret data completely. In addition, the method can avoid intra-frame distortion drift and has better visual concealment.

Key Words: Key Words: H.264/AVC; reversible steganography; robust ; intra-frame distortion drift

0　引言

视频隐写是将秘密信息隐藏于视频中,非授权者无法感知其传递行为及内容的一种技术。 H.264/AVC (H.264)视频压缩标准因具有压缩率高、差错恢复能力强、适用范围广等优点已成为目前最流行的视频压缩标准,并在多个领域得到了迅速部署与广泛普及[1]。

隐藏了秘密信息的 H.264 视频在网络上传输时,因遭到恶劣的物理环境(无意攻击)、网络攻击（有意攻击）等而产生比特错、丢帧、丢包等，致视频损毁秘密信息无法恢复。2005 年学术界已经开始了 H.264 视频隐写的研究,但研究的文献大多数选择在变换域内,非常依赖于 DCT 系数[2~6]，针对视频隐写鲁棒性的算法研究依然很少[7]。Esen 等人[8]利用禁止区和选择嵌入的方法实现强鲁棒性视频隐写，Liu 等人[4~6]针对 H.264 视频利用 BCH 码来实现恢复比特错,但这些算法均不能实现丢帧、丢包等情况下秘密信息的恢复。

通常,秘密信息在嵌入和提取过程中会在一定范围内修改原始载体,这对那些如医疗、图像及法律证据等不容忍永久失真的载体来说是不可接受的。可逆隐写的出现可以解决这一问题。通常,把秘密信息在提取后能完全恢复原载体的隐写方法叫可逆隐写。目前,对于可逆隐写方法的研究大都集中在可逆性方面[10]，基于 H.264 视频可逆隐写算法的研究还不是很多[12]。文献[10,11]利用预测模式选择嵌入块实现信息隐藏，文献[10]只考虑算法的可逆性,文献[11]在文献[10]的基础上解决了比特位出错问题,提高了鲁棒性,但不能解决由于丢包、丢帧等引起的帧错、帧丢失等问题。文献[12]利用秘密共享实现鲁棒性,但是算法不能解决比特位出错、丢失等问题。文献[13]利用直方图实现了基于 3D H.264/AVC 的可逆视频隐写。

本文针对 H.264 视频,提出了一种新的可逆鲁棒视频隐写方法。该方法在信息提取完全恢复原视频的同时,可以恢复由于恶

基金项目：国家自然科学基金资助项目（61272407）

作者简介：刘舒扬（1998-），男，河南周口人，本科生，主要研究方向为信息安全（liushy2016@lzu.edu.cn）；陈亮（1983-），男，江西吉安人，高级工程师，博士，主要研究方向为信息安全.

劣物理环境、网络攻击等引起的比特丢失、比特错、帧丢失及帧错误等,有并能效控制帧内失真漂移,视觉效果良好。

1　基本概念

1.1　帧内失真漂移预防

H.264 视频压缩编码标准中 4×4 块的帧内预测模式有 9 种,16×16 块的帧内预测模式有 4 种,分别用 0~8 和 0~3 表示。在帧内预测时,预测块是依据已编码块和待编码的当前块（当前块）来计算的。如图 1 所示,假定当前块为,则 $B_{i,j}$ 的预测值是由其已编码的左邻块 $B_{i,j-1}$、左上邻块 $B_{i-1,j-1}$、上邻块 $B_{i-1,j}$ 及右上邻块 $B_{i-1,j+1}$ 块中灰色部分,根据当前块所采用的帧内预测模式计算出来。若在图中灰色部分嵌入秘密信息,则因嵌入所导致的误差必然会通过上述计算传递到当前块中,造成帧内失真漂移。若对当前块进行预测时,不采用嵌入秘密信息的邻块边缘像素,帧内失真漂移就能预防。

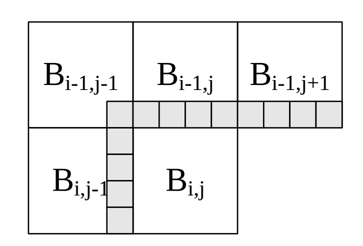


图 1　帧内预测块及其邻块

为使用方便,给出如下定义:

条件 1　右邻块 $\in \{0,3,7\}_{4\times4} \cup \{0\}_{16\times16}$ 。

条　件　2　左下邻块 $\in \{0,1,2,4,5,6,8\}_{4\times4} \cup \{0,1,2,3\}_{16\times16}$ 和下邻块 $\in \{0,8\}_{4\times4} \cup \{1\}_{16\times16}$ 。

条件 3　右下邻块 $\in \{0,1,2,3,7,8\}_{4\times4} \cup \{0,1,2,3\}_{16\times16}$ 。

当前块及三个条件如图 2 所示。

	当前块	预测模式: 4×4:0, 3, 7 16×16：0
预测模式: 4×4:0, 1, 2, 4, 5, 6, 8 16×16:0, 1, 2, 3	预测模式: 4×4:1, 8 16×16:1	预测模式: 4×4:0, 1, 2, 3, 7, 8 16×16:0, 1, 2, 3

图 2　当前块及三个条件

若当前块满足条件 1,则在该块内嵌入信息所导致的误差不会通过最右边一列传递到其右邻块中。

若当前块满足条件 2,则在该块内嵌入信息所导致的误差不会通过最下边一行传递到其左下与下邻块中。

若当前块满足条件 3,则在该块内嵌入信息所导致的误差不会通过最右下角的子块传递到其右下邻块。

因此,利用条件 1、2 和 3 可以控制帧内失真漂移。

1.2　BCH 码

若 n 为码长,k 为信息位,BCH(n, k, t)能纠正 t 个错误，则 BCH (n,k,t) 码的校验矩阵为

$$H=\begin{bmatrix}1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1}\\ 1 & \alpha^3 & (\alpha^3)^2 & \cdots & (\alpha^{n-1})^2\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ 1 & \alpha^{2t-1} & (\alpha^{2t-1})^2 & \cdots & (\alpha^{n-1})^2\end{bmatrix} \tag{1}$$

其中：α 是 $GF(2^m)$ 上的本原元。若原始二进制数据流为 $Q=\{q_0,q_1,\ldots,q_{n-1}\}$，要嵌入的秘密信息的数据流为 $V=\{v_0,v_1,\ldots,v_{n-1}\}$，则

$$V=QH^T \tag{2}$$

若经传输后接收到的码流数据为 $S=\{s_0,s_1,\ldots,s_{n-1}\}$，则 V 和 S 可分别表示为 $V(X)=v_0+v_1x+v_2x^2+v_3x^3+\cdots v_{n-1}x^{n-1}$ 与 $S(X)=s_0+s_1x+s_2x^2+s_3x^3+\cdots s_{n-1}x^{n-1}$,S 和 V 的差别 E 可表示为

$$S=V+E \tag{3}$$

由式（2）和（3）可得

$$Y=(S-H)H^T=EH^T \tag{4}$$

则嵌入秘密信息的 S 可由式(3)计算,秘密信息可由式(2)来提取.

1.3　秘密共享

秘密共享是一种将秘密分发给参与者,至少有若干个参与者才能将秘密恢复的技术。1979 年 Shamir[9]提出了基于拉格朗日插值的门限秘密共享方法。

假设要分割的秘密信息为 $a_0=k$,构造 t-1 次方多项式 $p_{t-1}(x)=a_0+a_1x+a_2x^2+\cdots a_{t-1}x^{t-1} \bmod p$ 。其中，大素数 $p>0$，$a_i\in Z_p\,(i=1,\ldots,n)$。设有 $n(n>p)$ 个参与者,计算 t-1 次方多项式在 x_i 处的值 y_i ，将其作为子秘密 $(x_i,y_i)(i=1,\ldots,n)$ 分发给 n 个参与者。若多项式 $p_{t-1}(x_i)$ 的 t 个值为$(x_i,y_i)\,(i=1,\ldots,t)$,则由 Lagrangre 插值公式可得

$$p_{t-1}(x)=\sum_{i=1}^{t}y_i\prod_{j=1,j\neq i}^{t}\frac{x-x_j}{x_i-x_j} \tag{5}$$

则常数项为

$$k=p_{t-1}(0)=\sum_{i=1}^{t}y_i\prod_{j=1,j\neq i}^{t}\frac{-x_j}{x_i-x_j}=\sum_{i=1}^{t}b_iy_i\quad(b_i=\prod_{j=1,j\neq i}^{t}\frac{-x_j}{x_i-x_j})$$

即为要恢复的秘密消息.

2　算法嵌入与提取过程

2.1　嵌入过程

由于 16×16 块的视频内容相对平缓,嵌入秘密信息容易引起视觉失真,所以本算法选用量化后的 4×4 块 DCT 系数来嵌入信息。图 3 给出了算法嵌入部分的操作流程。首先对 H.264 原始视频进行熵解码,得到解码后的 DCT 系数和 4×4 的帧内亮度预测模式。根据 DCT 系数中 DC 绝对值及自定义参数

threshold,选取合适的 4×4 块，根据当前块是否同时满足条件 1、2 和 3 来选择嵌入块。对已 BCH 编码的子秘密进行嵌入操作，最后对所有的已嵌入已编码信息的 DCT 系数重新熵编码得到嵌入秘密信息的目标视频。

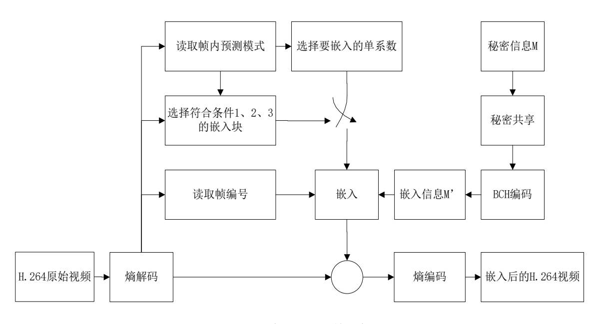


图 3　嵌入操作流程

以正整数 N 与系数 $\tilde{Y}_{ij}$（i,j=0,1,2,3 ）为例来具体描述算法嵌入过程，步骤如下：

a）按秘密共享分发子秘密。

b）对子秘密进行 BCH 编码。

c）根据直流系数的绝对值和自定义参数 threshold 的值选择可嵌入块（因为有非零系数的块进行调制时引起的失真效果不明显,所以选择具有非零系数及 $\tilde{Y}_{00}\geq 0$ 的块作为备选嵌入块）。

d）根据 c）中选择可嵌入块是否同时满足条件 1、2 和 3 选择嵌入块。依据调制方法把经过 BCH 编码的子秘密嵌入到满足条件的 DCT 系数中。

e）重新对 DCT 系数进行熵编码得到 H.264 目标视频。

调制方法：

如果 $|\tilde{Y}_{ij}|=N+1$ 或 $|\tilde{Y}_{ij}|\neq N$, $\tilde{Y}_{ij}$ 按式 (6)进行修改。

如果嵌入的比特是 1 且 $|\tilde{Y}_{ij}|=N$, $\tilde{Y}_{ij}$ 式(7)进行修改。

如果嵌入的比特是 0 且 $|\tilde{Y}_{ij}|=N$, $\tilde{Y}_{ij}$ 不做修改。

$$\tilde{Y}_{i,j}=\begin{cases}\tilde{Y}_{i,j}+1, & \text{if } \tilde{Y}_{i,j}\geq 0 \text{ and } |\tilde{Y}_{i,j}|=N+1,\\ \tilde{Y}_{i,j}-1, & \text{if } \tilde{Y}_{i,j}<0 \text{ and } |\tilde{Y}_{i,j}|=N+1,\\ \tilde{Y}_{i,j}, & \text{if } |\tilde{Y}_{ij}|\neq N.\end{cases} \tag{6}$$

$$\tilde{Y}_{i,j}=\begin{cases}\tilde{Y}_{i,j}+1, & \text{if } \tilde{Y}_{i,j}\geq 0 \text{ and } |\tilde{Y}_{i,j}|=N,\\ \tilde{Y}_{i,j}-1, & \text{if } \tilde{Y}_{i,j}<0 \text{ and } |\tilde{Y}_{i,j}|=N.\end{cases} \tag{7}$$

2.2　提取过程

图 4 给出了算法提取的操作流程。首先对接收到的 H.264 视频进行熵解码操作,得到解码后 DCT 系数和 4×4 的帧内预测模式。根据 DCT 系数中 DC 绝对值、自定义阈值 threshold 以及邻块的预测模式是否同时满足条件 1、2 和 3,选择已嵌入秘密信息的 4×4 块，并从其量化 DCT 系数中提取已编码的秘密信息 M。

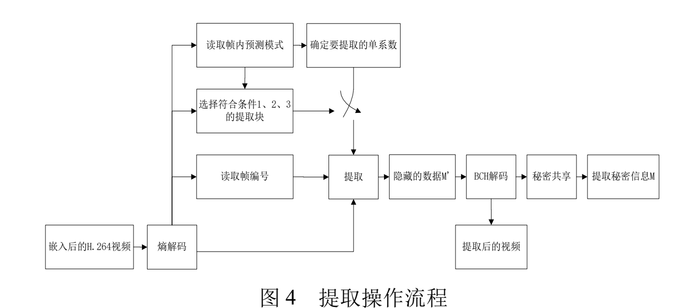


图 4　提取操作流程

提取方法：

如果 $|\tilde{Y}_{ij}|=N+2$ 或 $|\tilde{Y}_{ij}|\neq N+2$ 或 $|\tilde{Y}_{ij}|\neq N+1$, $\tilde{Y}_{ij}$ 按式(8)进行修改。

如果 $|\tilde{Y}_{ij}|=N+1$,提取比特位 1 且 $\tilde{Y}_{ij}$ 按式(9)进行修改。

如果 $|\tilde{Y}_{ij}|=N$,提取的比特位 0 且 $\tilde{Y}_{ij}$ 不做修改。

$$\tilde{Y}_{i,j}=\begin{cases}\tilde{Y}_{i,j}-1, & \text{if } \tilde{Y}_{i,j}\geq 0 \text{ and } |\tilde{Y}_{i,j}|=N+2,\\ \tilde{Y}_{i,j}+1, & \text{if } \tilde{Y}_{i,j}<0 \text{ and } |\tilde{Y}_{i,j}|=N+2,\\ \tilde{Y}_{i,j}, & \text{if } |\tilde{Y}_{i,j}|\neq N+2 \text{ or } |\tilde{Y}_{i,j}|\neq N+1 .\end{cases} \tag{8}$$

$$\tilde{Y}_{i,j}=\begin{cases}\tilde{Y}_{i,j}-1, & \text{if } \tilde{Y}_{i,j}\geq 0 \text{ and } |\tilde{Y}_{i,j}|=N+1,\\ \tilde{Y}_{i,j}+1, & \text{if } \tilde{Y}_{i,j}<0 \text{ and } |\tilde{Y}_{i,j}|=N+1.\end{cases} \tag{9}$$

提取信息 M′ 后,对其进行 BCH 解码和秘密共享处理即可得到秘密信息 M。

3　实验分析

本方法在 H.264 标准编/解码软件 JM 16.0 上进行了实现.每个实验视频以 30 bps 编码 300 个帧,I 帧编码间隔为 15,量化参数为28,分辨率为176×144。实验视频序列为container、news、coastguard、mobile 等标准视频.其中,PSNR (peak signal to noise ratio）峰值信噪比是由未编码的 YUV 视频文件与嵌入视频文件计算得到且比值为 I、B 与 P 帧的 PSNR 的平均值。原始视频是指秘密信息嵌入之前的 H.264 视频文件。丢帧率是所测试中丢失的帧数目占测试帧数目的之比；存活率是正确提取的嵌入比特数目与总嵌入比特数目之比。

表 1 给出了采用 BCH(63,7,15)、(t,n）门限秘密共享为(3,8)时,实验视频在不同的丢帧率下嵌入秘密信息的恢复情况。由表可知，4 个视频的平均信噪比为 36.71 dB,平均嵌入容量为 7 812 bit,在随机丢帧率为 5%、10%和 15%的情况下,嵌入秘密信息的存活率几乎达到 100%，可见算法具有较好的视觉质量、嵌入容量及较强的鲁棒性。在正常的网络丢包率（10%）下,完全适用于秘密信息的安全传输。此外,可以通过调节秘密共享门限值来调整嵌入容量以满足所要嵌入秘密信息,还可以通过使用视频有无限多帧这一特点来满足所需嵌入的秘密信息。

表 2 为实验视频 container 在网络丢帧率为 15%时,不同的(t,n)门限值的嵌入性能比较。由表 2 可知,当 t 保持不变，n 值变大时,存活率变大；当 n 保持不变,t 值变大时,存活率变小。这与

秘密共享的理论分析是一致的。 图 5 和 6 再次验证了这个结论。图 5 所示为 t 值不变 n 值变化时,不同丢帧率下秘密信息存活率的变化情况。图 6 为 n 值不变 t 值变化时,不同的丢帧率下秘密信息的存活率的变化情况。

图 7~11 为不同网络丢帧率下,不同实验次数下的嵌入秘密信息的存活率变化情况。由图可知,在网络丢帧率不大于 10%的情况下,所有实验下的嵌入比特存活率都达到了 100%。

表 1　不同丢帧率的性能比较

视频	PNS1 (dB)	嵌入容量（bit）	比特增加率/%	网络丢帧率/%	存活率/%
container	36.57	7 560	0.025	5	100.00
				10	100.00
				15	100.00
				20	92.31
news	37.08	8 064	0.025	5	100.00
				10	100.00
				15	92.86
				20	100.00
mobile	34.93	8 064	0.024	5	100.00
				10	100.00
				15	100.00
				20	64.29
coastguard	38.26	7 560	0.036	5	100.00
				10	100.00
				15	100.00
				20	100.00

表 2　不同(t,n)值的性能比较

（t,n）	嵌入容量（bit）	PSNR1（dB）	存活率/%
(2,8)	82	36.51	100.00
(3,8)	82	36.51	100.00
(4,8)	82	36.48	78.54
(3,4)	165	36.50	55.63
(3,16)	35	36.67	100.00
(3,32)	16	36.72	100.00

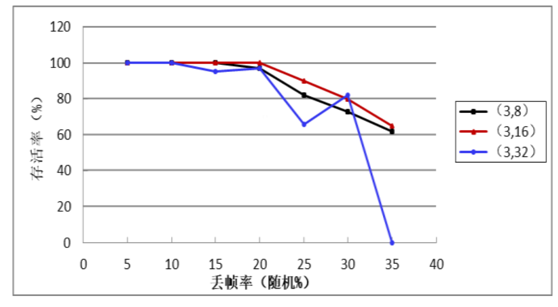


图 5　t 值不变 n 值变化时的存活率

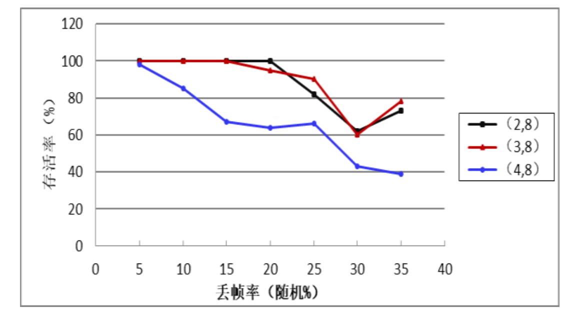


图 6　n 值不变 t 值变化时的存活率

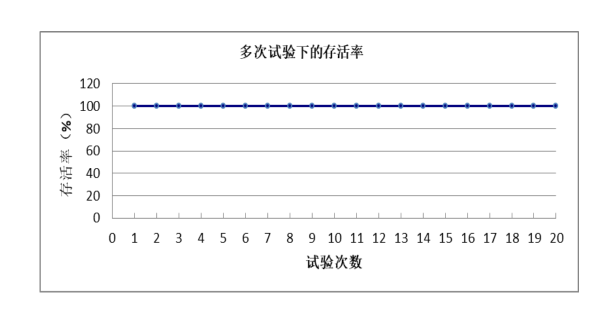


图 7　丢帧率 5%时多次实验的存活率

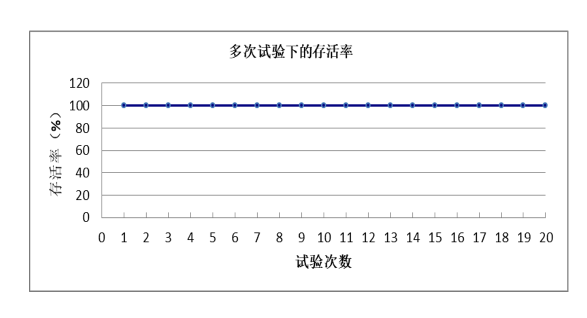


图 8　丢帧率 10%时多次实验的存活率

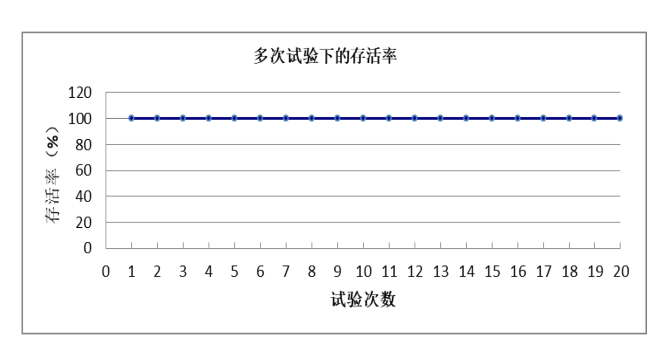


图 9　丢帧率 15%时多次实验的存活率

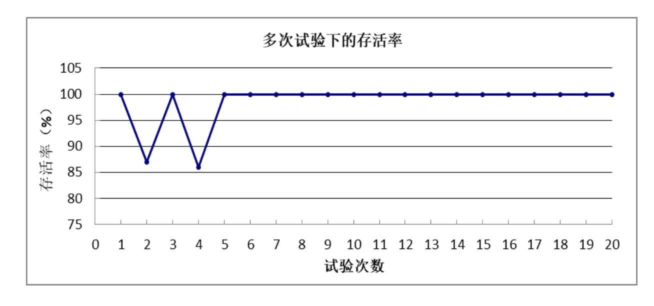


图 10　丢帧率 15%时多次实验下的存活率

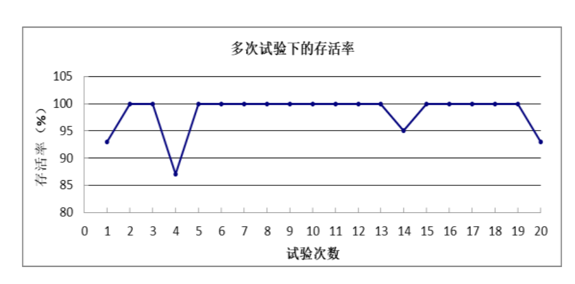


图 11　丢帧率 20%时多次实验的存活率

图 12~14 给出了 News 原视频,News 嵌入视频 及 News 提取信息后的视频帧。由图可以看出本算法具有较好的视觉效果。



图 12　原视频帧



图 13　嵌入视频帧



图 14　提取后的视频帧

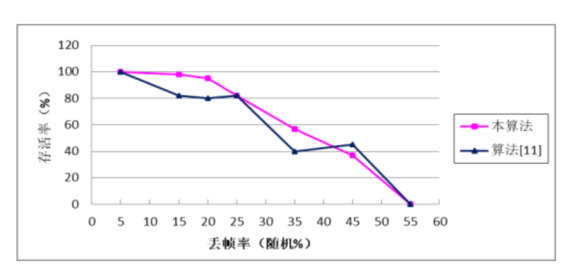


图 15　本算法与算法[11]对比

图 15 以视频 NEWS 为例给出了本算法与算法[11]的性能对比。由图可知,本算法的存活率达到 100%,本算法明显优于算法[11].这是因为算法[11]失去了 BCH 码的比特位纠错能力。

表 3 给出了本算法与算法[2,3]的性能对比。由表可知,在网络丢帧率为 5%、15%、20%,的情况下,算法[2]、算法[3]及本算法的平均存活率分别为 83.25 %、76.13 %、97.63 %。在网络丢帧率 15%的情况下,在正常网络传输的情况下,本文所提出的视频隐写算法能完全恢复秘密信息,且具有较高的鲁棒性。

表 3　本算法与其他算法的性能对比

视频	网络丢帧率（随机%）	算法[2]存活率/%	算法[3]存活率/%	本算法存活率/%
container	15	86.10	83.27	100.00
	20	79.23	72.73	92.31
news	15	87.20	82.14	92.86
	20	79.97	72.67	100.00
mobile	15	87.50	80.07	100.00
	20	78.90	72.30	94.29
coastguard	15	87.42	76.75	100.00
	20	79.68	69.12	100.00

4　结束语

本文提出了一种基于 H.264 的可逆鲁棒视频隐写方法.实现载有秘密信息的 H.264 视频在传输过程中遭到一种或多种有意或是无意的攻击导致视频损毁,依然可以在接收端恢复原始秘密信息,同时可以恢复原视频载体,可以应用于远程医疗、云计算、视频传输、可逆图像认证和二维向量地图等。由于秘密共享与 BCH 码技术都是在嵌入前与提取后使用，所以算法具有较小的复杂度。若 m1 是原始秘密信息比特位的个数，m2 是原始秘密信息的个数，则算法的嵌入与提取过程的复杂度都为 O(logm1+ m2)，能适应视频的实时性要求。

参考文献：

[1] 毕厚杰. 新一代视频压缩编码标准 H. 264//AVC [M]. 北京：人民邮电出版社, 2005: 97-129.

[2] Ma X J, Li Z T, Tu H, et al. A data hiding algorithm for H. 264//AVC video streams without intraframe distortion drift [J]. Circuits and Systems for Video Technology, 2010, 20 (10): 1320-1330.

[3] Ma X J, Li Z T, Lvy J, et al. Data hiding in H. 264//AVC streams with limited intra-frame distortion drift [C]// Proc of Computer network and Multimedia Technology. 2009: 1-5.

[4] Liu Y X, Li Z T, Ma X J, et al. A robust without intra-frame distortion drift data hiding algorithm based on H. 264//AVC [J]. Multimedia Tools & Applications, 2014, 72 (1): 613-636.

[5] Liu S, Liu Y X, Feng C, et al. A reversible data hiding method based on HEVC without distortion drift [C]// Proc of International Conference on Intelligent Computing. 2017: 613-624.

[6] Liu Y X, Liu S Y, Zhao H G, et al. A data hiding method for H. 265 without intra-frame distortion drift [C]// Proc of International Conference on

Intelligent Computing, Intelligent Computing Methodologies. 2017: 642-650.

[7] Liu Y X, Li Z T, Ma X J, et al. A robust data hiding algorithm for H. 264//AVC video streams [J]. Journal of Systems and Software, 2013, 86 (8): 2174-2183.

[8] Esen E, Alatan A A. Robust video data hiding using forbidden zone data hiding and selective embedding [J]. IEEE Trans on Circuits and Systems for Video Technology, 2011, 21 (8): 1130-1138.

[9] Shamir A. How to share a secret [J]. Communications of the ACM, 1979, 22 (11): 612-613.

[10] Liu Y X, Ma X J, Li Z T. A reversible data hiding scheme based on H. 264//AVC without distortion drift [J]. Journal of Software, 2012, 7 (5): 1059-1065.

[11] Liu Y X, Ju L M, Hu M S, et al. A robust reversible data hiding scheme for H. 264 without distortion drift [J]. Neurocomputing, 2015, 151 (1): 1053-1062.

[12] Liu Y X, Ju L M, Hu M S, et al. A new data hiding method for H. 264 based on secret sharing [J]. Neurocomputing, 2016, 188: 113-119.

[13] Zhao J, Li Z T, Feng B. A novel two-dimensional histogram modification for reversible data embedding into stereo H. 264 video [J]. Multimedia Tools & Applications, 2016, 75 (10): 5959-5980.